Apple IIGS
#103: Inline Procedure Name Format

Modified by: Matt Deatherage May 1992
Written by: Dave Lyons December 1991

This Technical Note describes a simple format for imbedding procedure names in object code, for use by debugging utilities.

Changes since December 1991: Changed `&syscnt` to `&SYSCNT` so it works with the `CASE ON` APW directive. Clarified the possible addition of parameters after the Pascal string.

GSBug 1.5b18 and later support a simple convention for including procedure names inline in the object code, for debugging purposes.

Inline Name Format

```
82 xx xx                          brl pastName
71 77                             dc.w $7771
nn xx xx xx xx...                 str 'the name string'
                       pastName   ...
```

That is, an imbedded name is a `BRL` around a signature word and a Pascal string. The name string can theoretically be up to 255 characters long, but in practice only short names are useful. For example, GSBug displays only the first 15 characters of a name when it is encountered, and only the first 11 when it appears as the operand of a `JSR` or `JSL` instruction.

Names in this format always start with a `BRL`, not a `BRA` or `JMP`. Signature word values other than $7771 are **reserved** for future definition, and more information may be added after the Pascal string.

Be careful what you name!

Be careful not to name something important—like a table, or a label from which you compute other addresses. The extra bytes generated by the inline name would mess up your calculations. If you name a heartbeat task, out-of-memory queue routine, or other construction that needs a special header, be sure to put the name where the executable code starts, not at the beginning of the header.

APW Assembly Macro

The following macro is for the APW assembler. If you equate `DebugSymbols` to zero, the macro generates no object code. If `DebugSymbols` is nonzero, the macro generates an inline name corresponding to its label.

Use the `name` macro anywhere you would use a label. For example:

```
DebugSymbols   GEQU 1
...
CountItems     name
```

The macro:

```
     MACRO
&lab name
&lab anop
     aif DebugSymbols=0,.pastName
     brl pastName&SYSCNT
     dc i'$7771'
     dc i1'L:&lab',c'&lab'
pastName&SYSCNT anop
.pastName
     MEND
```

MPW IIGS Assembly Macros

The following macros are for the MPW IIGS assembler. If you equate `DebugSymbols` to zero, the macros generate no object code. If `DebugSymbols` is nonzero, the macros generate inline names corresponding to their labels.

Use the `name` macro anywhere you would use a label. Use the `procname` macro in place of a `proc` directive, at the beginning of a procedure. For example:

```
DebugSymbols      equ 1
...
CountItems        name
TaskLoop          procname
```

The macros:

```
          macro
&lab      name
&lab
          if DebugSymbols<>0 then
          brl @pastName
          lclc &olds
&olds     setc &setting('string')
          string asis
          dc.w $7771
          dc.b &len(&lab),'&lab'
          string &olds
@pastName
          endif
          mend

* You can use procname instead of proc

          macro
&lab      procname   &x
&lab      proc       &x
          if DebugSymbols<>0 then
          brl @pastName
          lclc &olds
&olds     setc &setting('string')
          string asis
          dc.w $7771
```

```
            dc.b &len(&lab),'&lab'
            string &olds
@pastName
            endif
            mend
```

Writing utilities that recognize inline names

If you write a utility that recognizes inline procedure names in this format, check for a signature word of $777x, not specifically $7771. This allows more information to be added to the format later (a signature of $7772 could mean there is a Pascal string followed by parameter-passing information, for example).